

THIS OPINION WAS NOT WRITTEN FOR PUBLICATION

The opinion in support of the decision being entered today  
(1) was not written for publication in a law journal and  
(2) is not binding precedent of the Board.

Paper No. 15

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES

---

Ex parte WESLEY D. MCCULLOUGH  
and ROHIT A. VIDWANS

---

Appeal No. 95-4708  
Application 08/127,782<sup>1</sup>

---

ON BRIEF

---

Before HAIRSTON, JERRY SMITH and LEE, Administrative Patent Judges.

LEE, Administrative Patent Judge.

**DECISION ON APPEAL**

This is a decision on appeal under 35 U.S.C. § 134 from the final rejection of claims 1-56. No claim has been allowed.

**References relied on by the Examiner**

Hinton et al. (Hinton) Statutory Invention Registration H1291  
February 1, 1994 (filed Dec. 20, 1990)

Mike Johnson (Johnson), "Superscalar Microprocessor Design,"  
Prentice Hall Publication, pp. 103-126, 1991.

---

<sup>1</sup> Application for patent filed September 27, 1993.

### **The Rejection on Appeal**

Claims 1-56 stand finally rejected under 35 U.S.C. § 103 as being unpatentable over Hinton and Johnson (Paper No. 5, pages 2 and 4).

### **The Invention**

The invention is directed to an apparatus and method for storing results of an executed set of operations into a register file. In particular, the individual operations target the same register or the same portion of a register. Based in part on a prioritizing scheme, the operation results are written into the register file within one clock cycle. In that regard, however, claim 1 recites one half clock cycle rather than one clock cycle.

The independent claims are claims 1, 10, 19, 27, 35, 43, 51 and 54. Representative claim 19 is reproduced below:

19. An apparatus for storing results of multiple executed uops into a register file within one clock cycle, said uops executed by a superscalar microprocessor, said register file having a plurality of registers, said apparatus comprising:

memory logic for receiving names of a first destination register and a second destination register, said first destination register targeted by a first uop and said second destination register larger than said first destination register and targeted by a second uop;

merging logic for generating an enable signal for said second uop that corresponds to said first destination register if said second destination register includes said first destination register;

priority logic for asserting a write enable signal corresponding to said first destination register for a highest priority uop between said first and said second uop, if said first and said second uop have enable signals corresponding to said first destination register; and

enable logic for steering data associated with said highest priority uop from said memory logic to said first destination register of said register file according to said write enable signal within said one clock cycle.

### **Opinion**

We do not sustain the rejection of claims 1-56 under 35 U.S.C. § 103 as being unpatentable over Hinton and Johnson.

Each of the independent claims 1, 10, 19, 27, 35, 43, 51 and 54, in one form or other, requires the results of operations targeting the same register, in whole, part or portion, or corresponding enable signals, to be prioritized such that the results are written into the commonly targeted area within one clock cycle according to that priority. In the context of the appellants' disclosure, the writing of plural results into the same targeted area in the same clock cycle according to a determined priority does not mean that each of the results is actually written in the same clock cycle. Rather, the writing of that result which would become overwritten in the same clock cycle if the operations are orderly executed is given a lower priority and thus omitted, skipped, or ignored. See the specification from page 15 to page 18. The end result achieved

Appeal No. 95-4708  
Application 08/127,782

at the end of the clock cycle is as if all operations targeting the same area were performed. That is the proper interpretation. Neither the appellants nor the examiner urges a different view.

The issues on appeal center about whether Hinton and Johnson discloses or suggests, whether alone or in combination with each other, the writing of results into the same targeted register part or portion within the same clock cycle, a feature which the examiner has not denied is required by all the independent claims. The appellants argue that they do not. We agree.

With regard to the register file RF 6 of Hinton, the examiner stated (answer at 3):

The register file (RF) 6 is multi-ported, including two write ports. The RF can receive the results of two operations simultaneously. The REG coprocessors 10 can write the results of an arithmetic operation to the RF 6, simultaneously with the MEM coprocessors 10 loading an operand to the RF 6 from external memory. In other words, results of concurrently executing micro-operations can be written to the RF 6 simultaneously.

Also, in response to the appellants' argument, the examiner pointed to a portion of Hinton which defines two access ports for the register file 6 which (column 12, lines 64-66) "allow LOAD data from a previous read operation and STORE data from a current write access to be processed in the register simultaneously."

The problem with the examiner's position is that Hinton evidently is discussing simultaneous access to the register file

6 which is a 36 entry by 32-bit register file, not to the same entry or portion of any one entry in the register file. While one result is being stored in one entry, a different one can be read from another. There is no teaching or suggestion from Hinton that the same register parts, portions, or areas can or should be accessed simultaneously in one clock cycle.

The appellants are correct that Hinton desires to avoid conflicting access to the same register areas. In column 7, lines 49-50, Hinton states: "Hardware checks for dependencies and only issues the instructions that can be executed." In column 8, lines 56-68, Hinton states:

During the second pipe stage shown in FIG. 3, the resources [such as a register] are checked concurrently with the issuing and beginning of the instructions so this does not slow down the operating frequency. Each instruction is conditionally canceled and resued [sic, reissued] depending on the resource check for that instruction. Register Scoreboarding sets the destination register or registers busy once it passes the resource check. When the result returns -- whether 1 or many cycles later -- the resultant register gets marked as not busy and free to use. Each multicycle functional unit maintains a busy signal that is used to delay a new instruction that needs to use this busy unit.

Also, in column 12, lines 28-34, Hinton states:

A subsequent operation needing that specific register resource will be delayed until this long operation is completed. This is called scoreboarding the register. There is one bit per 32-bit register called the

scoreboard bit that is used to mark it busy if a long instruction. This scoreboard bit is checked during q12.

The appellants correctly argue (Br. at 6) that Hinton's solution to conflicting access to the same register area is to delay the issuance of one of the operations to eliminate the conflict. The appellants are correct (Br. at 6) that Hinton's scheme "fails to allow a register file update of two or more operations targeting the same register (or portion thereof) within a single clock cycle as allowed by the present invention as claimed." In other words, no writing of results is effectively carried out by being omitted, ignored, or deleted.

Further in support of their argument, the appellants point out (Reply at 2) that Hinton indicates (column 5, lines 45-60) that its register file 6 is more particularly described in patent application 07/486,407 (now Patent No. 5,185,872 to Arnold et al.). The appellants refer (Reply at 6) to the following description in Arnold et al. (Column 5, line 65, to column 6, line 5):

Since the both register and memory types of instructions allowed to execute in the same cycle, six possible register requests could be executing. Thus, a 6-port register file design is required to correctly implement these parallel functions. Of course, a mechanism must exist that prevents the collision of data, since writing the same register from multiple sources could be disastrous. To protect against this

problem, and to prevent data from being read before it is properly written, the RF [register file] uses register scoreboarding.

The above-quoted description from Arnold et al. does further support the appellants' reading of Hinton.

We agree with the appellants that nowhere does Hinton describe or suggest that two writing operations to the same register or register part or portion, are "effectively" processed during the same clock cycle. Hinton allows reads and writes to different registers in the register file to occur but not to the same targeted register or register parts. See Hinton in column 2, lines 45-53. Hinton teaches that in case of conflict, one of the operations will be canceled and reissued at a later time. See column 8, lines 56-68.

Johnson does not make up for the above-mentioned deficiencies of Hinton. As is correctly noted by the appellants (Br. at 9), Johnson discloses that operations that have been executed (but not yet allowed to update the register file) are placed in a reorder buffer. From the reorder buffer, the operations are allowed to update the register file "program code order." The examiner relied on Johnson for the teaching of an arbitration scheme based on program code order (answer at 5, lines 8-10). However, what is missing from Hinton is the idea of

"effectively" writing into the same targeted register areas in the same clock cycle (one of the writing is not just delayed to be processed at another time), not a different arbitration scheme which puts the conflicting operations in another order.

The appellants correctly argue (Br. at 9) that Johnson does not teach or suggest that multiple operations can update the same destination register (or portion thereof) within a common clock cycle. The appellants further correctly note (Br. at 9-10) that in its section 6.1.2, Johnson teaches that results from the reorder buffer are written into the register file "in sequential order." The examiner has failed to demonstrate how Johnson would reasonably suggest writing into the same register areas in the same clock cycle.

For the foregoing reasons, neither Hinton nor Johnson reasonably would have suggested writing into the same register parts in the same clock cycle. We also see no reason why or how their combination would have suggested writing into the same register parts in the same clock cycle. Accordingly, the rejection of claims 1-56 under 35 U.S.C. § 103 as being unpatentable over Hinton and Johnson cannot be sustained.



Appeal No. 95-4708  
Application 08/127,782

Conclusion

The rejection of claims 1-56 under 35 U.S.C. § 103 as being unpatentable over Hinton and Johnson is reversed.

**REVERSED**

KENNETH W. HAIRSTON	)	
Administrative Patent Judge	)	
	)	
	)	
	)	BOARD OF PATENT
JERRY SMITH	)	
Administrative Patent Judge	)	APPEALS AND
	)	
	)	INTERFERENCES
	)	
JAMESON LEE	)	
Administrative Patent Judge	)	

Appeal No. 95-4708  
Application 08/127,782

Blakely, Sokoloff, Taylor & Zafman  
12400 Wilshire Boulevard  
7th Floor  
Los Angeles, CA 90025